Ensuring Confidentiality in Process Mining

Majid Rafiei[1][0000−0001−7161−6927], Leopold von Waldthausen[2][0000−0001−8892−8361], and Wil M.P. van der Aalst[1][0000−0002−0955−6940]

[1] Chair of Process and Data Science, RWTH Aachen University, Aachen, Germany
majid.rafiei@pads.rwth-aachen.de
http://www.pads.rwth-aachen.de/go/id/pnbx/?lidx=1
[2] Yale University, New Haven, USA

Abstract. To gain novel and valuable insights into the actual processes executed within a company, process mining provides a variety of powerful data-driven analyses techniques ranging from automatically discovering process models to detecting and predicting bottlenecks, and process deviations. On the one hand, recent breakthroughs in process mining resulted in powerful techniques, encouraging organizations and business owners to improve their processes through process mining. On the other hand, there are great concerns about the use of highly sensitive event data. Within an organization, it often suffices that analysts only see the aggregated process mining results without being able to inspect individual cases, events, and persons. When analysis is outsourced also the results need to be encrypted to avoid confidentiality problems. Surprisingly, little research has been done toward security methods and encryption techniques for process mining. Therefore, in this paper, we introduce a novel approach that allows us to hide confidential information in a controlled manner while ensuring that the desired process mining results can still be obtained. We provide a sample solution for process discovery and evaluate it by applying a case study on a real-life event log.

Keywords: Responsible process mining · Confidentiality · Process discovery · Directly follows graph

1 Introduction

Data science is changing the way we do business, socialize, conduct research, and govern society. Data are collected on anything, at any time, and in any place. Therefore, it is not surprising that many are concerned about the usage of such data. The Responsible Data Science (RDS) [6] initiative focuses on four main questions: (1) Data science without prejudice (How to avoid unfair conclusions even if they are true?), (2) Data science without guesswork (How to answer questions with a guaranteed level of accuracy?), (3) Data science that ensures confidentiality (How to answer questions without revealing secrets?), and (4) Data science that provides transparency (How to clarify answers such that they become indisputable?). This paper focuses on the confidentiality problem (third question) when applying process mining to event data.

In recent years, process mining has emerged as a new field which bridges the gap between data science and process science. Process mining uses event data to provide novel insights [1]. The breakthroughs in process mining are truly remarkable. Currently, over 25 commercial tools supporting process mining are available (e.g., Celonis, Disco, Magnaview, QPR, etc.) illustrating the value of event data [4]. However, existing tools and also the corresponding research rarely considers confidentiality issues. Since the event logs used as a basis for process mining often contain highly sensitive data, confidentiality is a major problem.

As we show in this paper, *confidentiality in process mining cannot be achieved by simply encrypting all data*. Since people need to use and see process mining results, the challenge is to retain as little information as possible while still being able to have the same desired result. Here, the desired result is a process model that can be used to check compliance and spot bottlenecks. The discovered models based on encrypted event logs should be identical to the results obtained for the original event data (assuming proper authorizations).

In this paper, we present a new approach to deal with confidentiality in process mining. Selected parts of data will be encrypted or anonymized while also keeping parts of the original event logs. For example, activity names remain unchanged, but one cannot correlate events into end-to-end cases. The new approach is explained through a sample solution for process discovery based on a framework for confidentiality. The framework allows us to derive the same results from secure event logs when compared to the results from original event logs, while unauthorized persons cannot access confidential information. In addition, this framework provides a secure solution for process mining when processes are cross-organizational.

The remainder of this paper is organized as follows. Section 2 outlines related work and the problem background. In Section 3, we clarify process mining and cryptography as preliminaries. In Section 4, the problem is explained in detail. The new approach is introduced in Section 5. In Section 6 our evaluation is described, and Section 7 concludes the paper.

2 Related Work

In both data science and information systems, confidentiality has been a topic of interest in the last decade. In computer science, privacy-preserving algorithms and methods in differential privacy have the closest similarity to confidentiality in process mining. In sequential pattern mining, the field of data science most closely related to process mining, there has been work on preserving privacy in settings with distributed databases [9] or in cross-organizational settings [20].

The Process Mining Manifesto [5] also points out that privacy concerns should be addressed. Although there have been a lot of breakthroughs in the field of process mining ranging from data preprocessing [18], and process discovery [16] to performance analysis [11], the research field confidentiality and privacy has received relatively little attention.

The topic of Responsible Process Mining (RPM) [2] has been put forward by several authors thereby raising concerns related to fairness, accuracy, confidentiality, and transparency. In [19] a method for securing event logs to be able to do process discovery by Alpha algorithm has been proposed. In [8] a possible approach toward a solution, allowing the outsourcing of process mining while ensuring the confidentiality of dataset and processes, has been presented. In [13] the authors has used a cross-organizational process discovery setting, where public process model fragments are shared as safe intermediates. There are also a few online guidelines [17].

3 Background

In this section, we briefly present the main concepts and refer the readers to relevant literature.

3.1 Process Mining

The four basic types of process mining are; (1) *process discovery*, which is used to learn a process model based on event data , (2) *conformance checking*, which compares observed behavior and modeled behavior , (3) *process reengineering*, used for improving or extending the process model , and (4) *operational support*, providing warning, predictions, and/or recommendations. In this paper, we focus on process discovery.

Events are the smallest data unit in process mining and occur when an activity in a process is executed. In Table 1 each row indicates an event with different attributes.

A trace is a sequence of events and represents for one instance how a process is executed. E.g., candidate George (case 3) is first registered, then admitted.

An event log is a collection of sequences of events. There are process mining algorithms that can use them as input. Event data are widely available in current information systems [5].

As you can see in Table 1, “Timestamp” identifies the moment in time at which an event has taken place, and “Case ID” is what all events in a trace have in common so that they can be identified as part of that process instance. Event

Table 1: Sample event log (each row represents an event).

Case ID	Event ID	Timestamp	Activity	Candidate	Cost
1	23	30-12-2010:11.02	Register	Peter	50
1	24	30-12-2010:12.08	Check documents	Peter	60
2	27	30-12-2010:13.16	Admit	Anna	150
2	26	30-12-2010:16.03	Register	Anna	50
1	25	30-12-2010:17.52	Admit	Peter	100
3	28	30-12-2010:17.57	Register	George	55
3	29	30-12-2010:18.19	Admit	George	145

logs can also include additional attributes for the events they record. There are two main attribute types that fall under this category. “Event Attributes” which are specific to an event, and “Case Attributes” which are ones that stay the same throughout an entire trace.

A Directly Follows Graph (DFG) is a graph where the nodes represent activities and the arcs represent causalities. Activities “a” and “b” are connected when “a” is frequently followed by “b”. The weights of the arrows denote the frequency of the relation [12]. Most commercial process mining tools use DFGs. Unlike more advanced process discovery techniques (e.g., implemented in ProM), DFGs can not express concurrency. The DFGs used in this paper also include times, i.e., besides the frequencies also the average time that it takes to go from one activity to another one is also included.

3.2 Cryptography

Cryptography or cryptology is about constructing and analyzing protocols that prevent third parties or the public from reading private messages [7].

Cryptosystem is a suite of cryptographic algorithms needed to implement a particular security service, most commonly for achieving confidentiality [10]. There are different kinds of cryptosystems. In this paper, we use the following ones.

– *Symmetric Cryptosystem:* In symmetric systems, the same secret key is used to encrypt and decrypt a message. Data manipulation in symmetric systems is faster than asymmetric systems as they generally use shorter key lengths. Advanced Encryption Standard (AES) is a symmetric encryption algorithm.
– *Asymmetric Cryptosystem:* Asymmetric systems use a public key to encrypt a message and a private key to decrypt it or vice versa. Use of asymmetric systems enhances the security of communication. Rivest-Shamir-Adleman (RSA) is an asymmetric encryption algorithm.
– *Deterministic Cryptosystem:* A deterministic cryptosystem is a cryptosystem which always produces the same ciphertext for a given plaintext and key, even over separate executions of the encryption algorithm.
– *Probabilistic Cryptosystem:* A probabilistic cryptosystem as opposed to deterministic cryptosystem is a cryptosystem which uses randomness in an encryption algorithm so that when encrypting the same plaintext several times it will produce different ciphertexts.
– *Homomorphic Cryptosystem:* A homomorphic cryptosystem allows computation on ciphertext. E.g. Paillier is a partially homomorphic cryptosystem.

4 Problem Definition

To illustrate the challenge of confidentiality in process mining, we start this section with an example. Consider Table 2 describing a totally encrypted event log, belonging to surgeries in a hospital. Since we need to preserve difference to find a sequence of activities for each case, discovering process model, and other analyses like social network discovery, "Case ID", "Activity", and "Resource" are encrypted based on a deterministic encryption method. Numerical data (i.e., "Timestamp" and "Cost") are encrypted by a homomorphic encryption method to be able to do basic mathematical computations. Now suppose that we have background knowledge about surgeons and the approximate cost of different types of surgeries and the question is whether this log is secure or not.

Owning to the fact that the "Cost" is encrypted by a homomorphic encryption method, the maximum value for the "Cost" is the real maximum cost and based on the background knowledge we know that e.g., the most expensive event in the hospital was the brain surgery by "Dr. Jone", on "01/09/2018 at 12:00", and the patient name is "Judy". Since "Case ID", "Activity", and "Resource" are encrypted by a deterministic encryption method, we can replace all these encrypted values with the corresponding plain values. Consequently, some part of the encrypted data could be made visible without decryption. This example clearly demonstrates that even when event logs are totally encrypted, given a small fraction of contextual knowledge, data leakage is possible.

There are also some other techniques, which can be used to extract knowledge from an encrypted event log, exploiting background knowledge and some specific characteristics of the event log. In the following, we describe some of them.

- *Exploring Order of Activities:* in large processes, most cases follow a unique path, which can cause data leakage by focusing on the order of activities [2].
- *Frequency Mining:* one can find the most or the less frequent activities and simply replace the encrypted values with the real values based on a knowledge about the frequency of activities.
- *Exploring Position of Activities:* limited information about the position of activities in traces can lead to data leakage. E.g., in a hospital, one can easily know that the first activity is registration.

These are just some examples to demonstrate that encryption alone is not a solution. For example, in [14] it is shown that mobility traces are easily identifiable after encryption. Any approach which is based on just encrypting the whole event log will have the following additional weaknesses:

- *Encrypted Results:* since results are encrypted, the data analyst is not able to interpret the results. E.g., as data analyst we want to know which paths are

Table 2: An entirely encrypted event log.

Case ID	Activity	Resource	Timestamp	Cost
1ab	Abc1dfg	0fgh14	123	5000
2cd	Chf5jkl	024sdfk	125	6000
3ty	215sfs0	.543s1s	254	3500
1tu	2154@3	3242s2	248	2000
1za	321$22	02315d	157	5500

the most frequent after "Registration" activity; how can we do this analysis when we do not know which activity is "Registration"? The only solution is decrypting results.
- *Impossibility of Accuracy Evaluation:* how can we make sure that a result of the encrypted event log is the same as the result of the plain event log? The only solution is decrypting the result of the encrypted event log.

Generally and as explored by [8], using cryptography is a resource consuming activity, and decryption is even much more resource consuming than encryption. These weaknesses demonstrate that it would be better if we could keep some parts of a data as plain text even in the secure event log. However, the challenge is to decide what should be kept in plain format and what not (encrypted or removed), and how we should address the data leakage that may arise from the plain data. In the next section, an approach is introduced, where we provide some answers to this questions.

5 Approach

As mentioned, the approach is described based on a sample solution for process discovery. In fact, the aim is to convert an event log to a secure event log such that just authorized persons can have access to confidential data, process model for the secure event log is the same as process model for the plain event log, and the current process discovery techniques can be used with the secure event log.

Fig. 1 shows the scheme which has been depicted as a framework to provide a solution for the above-mentioned purpose. This framework has been inspired by [4], where abstractions are introduced as intermediate results for relating models and logs. As can be seen in Fig. 1 three different environments and two confidentiality solutions are presented.

- *Forbidden Environment:* In this environment, the actual information system runs that needs to use the real data. The real event logs (EL) produced by this environment contain a lot of valuable confidential information and except some authorized persons no one can access this data.
- *Internal Environment:* This environment is just accessible by the authorized stakeholders. A data analyst can be considered as an authorized stakeholder and can access the internal event logs. Event logs in this environment are partially secure, selected results produced in this environment (e.g., a process model) are the same as the results produced in the forbidden environment, and data analyst is able to interpret the results without decryption.
- *External Environment:* In this environment, unauthorized external persons can access the data. Such environments may be used to provide the computing infrastructure dealing with large data sets (e.g., a cloud solution). Event logs in this environment are entirely secure, and the results are encrypted. Whenever data analyst wants to interpret the results, these results have to be decrypted and converted to the internal version. Also, results from the external environment do not need to be exactly the same as the results from the internal environment.

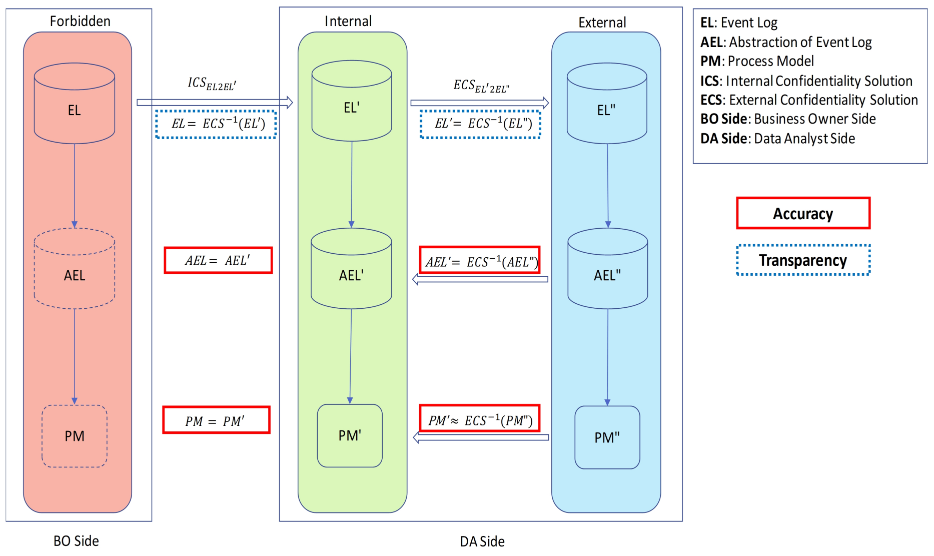


Fig. 1: The proposed framework for confidentiality in process mining.

5.1 Internal Confidentiality Solution (ICS)

For ICS we combine several methods and introduce the connector method, where several techniques are utilized to create a new level of security. Fig. 2 gives an overview of the anonymization steps.

Filtering and Modifying The Input. The first step to effective anonymization is preparing the data input. To filter the input, simple limits for frequencies can be set, and during loading an event log all traces that do not reach the minimal frequencies are not transferred to the EL'. Attributes which are irrelevant for analysis should be removed regardless of their sensitivity.

Choosing The Plain Data. As mentioned, we need to produce interpretable results. Hence, some parts of event log remain as plain text in the internal version of the secure event log (EL'). To make a process model based on EL', we should take a look at what information and/or structure is strictly necessary for discovering a process model. Here there are different choices; however, we consider the DFG, used by many discovery approaches, as an abstraction which relates logs and models [4]. Therefore, *Abstractions* (i.e., AEL, AEL', and AEL'') are DFGs.

If we have a DFG, then the process model can be made based on it. Therefore, the next step is taking a look at what information and/or structure is necessary to make a DFG. Since a DFG is a graph which shows the directly follows relation between activities, we need activities as information to be plain, and we also need

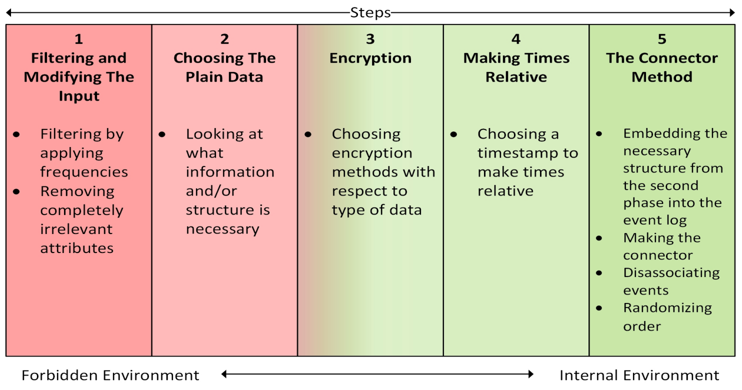

Steps
1
Filtering and Modifying The Input
Filtering by applying frequencies
Removing completely irrelevant attributes
2
Choosing The Plain Data
Looking at what information and/or structure is necessary
3
Encryption
Choosing encryption methods with respect to type of data
4
Making Times Relative
Choosing a timestamp to make times relative
5
The Connector Method
Embedding the necessary structure from the second phase into the event log
Making the connector
Disassociating events
Randomizing order
Forbidden Environment
Internal Environment

Fig. 2: The internal confidentiality solution.

a structure which can be used for extracting directly follows relations. Such a structure should be embedded into EL'.

Encryption. Here there are two important choices. The first choice is which columns of the event log should be encrypted. Second, we need to decide which algorithms should be used. As can be seen in Fig. 3, for the internal environment, we use Paillier as a good choice for numeric attributes (i.e. “Cost”) and AES for other attributes (i.e. “Activity”).

Making Times Relative. Times need to be modified because keeping the exact epoch time of an event can allow one to identify it. The naive approach, of setting the starting time of every trace to 0, would make it impossible to replay events and reconstruct the original log. Thus, we select another time that all events are made relative to. This time can be kept secure along with the keys for decryption. Fig. 3 shows a sample log after encrypting and making times relative to the “30.12.2010:00.00”.

The Connector Method. Using the connector method we embed the structure, which can be used for extracting directly follows relations, into EL'. Also, the connector method helps us to reconstruct the full original event logs when keys and relative values are given. In the first step, the previous activity (“Prev. Activity”) column is added in order of identifying which arcs can be directly added to the directly follows graph later.

In the second step, we find a way to securely save the information contained in the “Case ID”, without allowing it to link the events. This can be done by giving each row a random ID (“ID”) and a previous ID (“PrevID”). These uniquely identify the following event in a trace because the IDs are not generic like activity names. The ID for start activities is always a number of zeros. Fig. 4 shows the log after adding “Prev. Activity” and “PrevID”.

Case ID	Time stamp	Activity	Resource	Cost
1	30-12-2010:11.02	Registration	Peter	50
1	30-12-2010:12.08	IV Antibiotics	Maria	60
1	30-12-2010:13.16	Sepsis triage	Anna	100
2	30-12-2010:16.03	Registration	Peter	50
2	30-12-2010:17.52	Sepsis triage	Anna	150
3	30-12-2010:17.57	Registration	George	55
3	30-12-2010:18.19	Sepsis triage	Anna	145

(a) The sample event log.

Case ID	Time stamp	Activity	Resource	Cost
1	00-00-0000:11.02	Registration	ö?1	0820963
1	00-00-0000:12.08	IV Antibiotics	èÓäp	1450127
1	00-00-0000:13.16	Sepsis triage	õÀÕÒ	1850738
2	00-00-0000:16.03	Registration	ö?1	0820963
2	00-00-0000:17.52	Sepsis triage	õÀÕÒ	8385171
3	00-00-0000:17.57	Registration	ö+íMV7	0950822
3	00-00-0000:18.19	Sepsis triage	õÀÕÒ	5189192

(b) Encrypting resources and costs and making times relative.

Fig. 3: The event log after encrypting and making times relative

In the third step, regarding the fact that these columns contain the same information previously found in the "Case ID", they have to be hidden and secure. This can be done by concatenating the "ID" and "PrevID" of each row and encrypting those using AES. Due to the nature of AES neither orders nor sizes of the IDs can thus be inferred. The concatenation can be done in any style, in this example, we however simply concatenate the "PrevID" behind the "ID". To retain the "ID" and "PrevID" one simply needs to decrypt the "Connector" column and cut the resulting number in two equal parts. This method requires that every time the two IDs differ by a factor 10 a zero must be added to guarantee equal length. Fig. 5 shows the log after concatenating the ID columns and encrypting them as a connector.

In the final step, we use the "Case ID" to anonymize the "Time tamp". The "Time tamp" attribute of events which have the same "Case ID" is made relative to the preceded one. The exception is the first event of each trace which remains unchanged. This allows the complete calculation of all durations of the arcs in a directly follows graph but makes it complicated to identify events based on the epoch times they occurred at. After creating the relative times, we are free to delete the "Case ID" and randomize the order of all rows, ending up with an unconnected log in Fig. 6.

Case ID	Time stamp	Activity	Prev. Activity	Resource	Cost	ID	PrevID
1	00-00-0000:11.02	Registration	START	ö?1	0820963	23	00
1	00-00-0000:12.08	IV Antibiotics	Registration	èÓäp	1450127	28	23
1	00-00-0000:13.16	Sepsis triage	IV Antibiotics	õÀÕÒ	1850738	14	28
2	00-00-0000:16.03	Registration	START	ö?1	0820963	10	00
2	00-00-0000:17.52	Sepsis triage	Registration	õÀÕÒ	8385171	45	10
3	00-00-0000:17.57	Registration	START	ö+íMV7	0950822	21	00
3	00-00-0000:18.19	Sepsis triage	Registration	õÀÕÒ	5189192	04	21

Fig. 4: The event log after adding previous activities and previous IDs.

Case ID	Time stamp	Activity	Prev. Activity	Resource	Cost	Connector
1	00-00-0000:11.02	Registration	START	ö?1	0820963	2300
1	00-00-0000:12.08	IV Antibiotics	Registration	èÓäp	1450127	2823
1	00-00-0000:13.16	Sepsis triage	IV Antibiotics	õÀÕÒ	1850738	1428
2	00-00-0000:16.03	Registration	START	ö?1	0820963	1000
2	00-00-0000:17.52	Sepsis triage	Registration	õÀÕÒ	8385171	4510
3	00-00-0000:17.57	Registration	START	ö+íMV7	0950822	2100
3	00-00-0000:18.19	Sepsis triage	Registration	õÀÕÒ	5189192	0421

(a) Concatenating ID and previous ID.

Case ID	Time stamp	Activity	Prev. Activity	Resource	Cost	Connector
1	00-00-0000:11.02	Registration	START	ö?1	0820963	P/AzmbU
1	00-00-0000:12.08	IV Antibiotics	Registration	èÓäp	1450127	UpANzSf
1	00-00-0000:13.16	Sepsis triage	IV Antibiotics	õÀÕÒ	1850738	4wd1bmB
2	00-00-0000:16.03	Registration	START	ö?1	0820963	QcFOQZ
2	00-00-0000:17.52	Sepsis triage	Registration	õÀÕÒ	8385171	vqU70z3M
3	00-00-0000:17.57	Registration	START	ö+íMV7	0950822	F+ZjgkWS
3	00-00-0000:18.19	Sepsis triage	Registration	õÀÕÒ	5189192	G4om2+h

(b) Encrypting the connector.

Fig. 5: The event log after concatenating IDs and encrypting the connector

Fig. 6 is internally secure event log (EL'), which can be used by a data analyst to make DFG (AEL') and PM'. It is obvious that if process discovery could have been done on the plain event log (EL), AEL would be identical to AEL' (i.e., both of them are the same DFG) and PM would be identical to PM'.

Comparing Fig. 6 and the original log (Fig. 3a), one can see that there is no answer for the following questions in EL' anymore: (1) *What is the name of a resource?* (2) *Who was responsible for doing an activity at exact time t?* (3) *What is the sequence of activities which has been done for case c?* (4) *How long did it take to process case c?* (5) *What is the cost of activity a which has been done by resource r for case c?*

However, it is still possible to answer the following question: *Who is responsible for activity a?* In fact, EL' is a partially secure version of event log in such a way that contains the minimum level of information, which data analyst needs to reach the result. Although ICS does not preserve the standard format of the event log which is used by the current process discovery techniques, it provides an intermediate input (i.e., a DFG), which can be used by the current tools. In the External Confidentiality Solution (ECS), we need to avoid any form of data leakage, i.e., the results do not need to be interpreted by the external party.

Time	Activity	Prev. Activity	Resource	Cost	Connector
01.49	Sepsis triage	Registration	õÀÕÒ	8385171	vqU70z3M
17.57	Registration	START	ö+íMV7	0950822	F+ZjgkWS
01.08	Sepsis triage	IV Antibiotics	õÀÕÒ	1850738	4wd1bmB
16.03	Registration	START	ö?1	0820963	QcFOQZ
11.02	Registration	START	ö?1	0820963	P/AzmbU
01.06	IV Antibiotics	Registration	èÓäp	1450127	UpANzSf
01.22	Sepsis triage	Registration	õÀÕÒ	5189192	G4om2+h

Fig. 6: The output event log after applying ICS

5.2 External Confidentiality Solution (ECS)

In the external environment, the plain part of the event log may cause data leakage. E.g., based on background knowledge, one with a little effort can extract that who is responsible for "Registration". Therefore, in ECS, we convert El' to the externally secure event log (EL'') in such a way that it prevents an adversary from extracting valuable information even by inference. In the following, our two-steps ECS is explained.

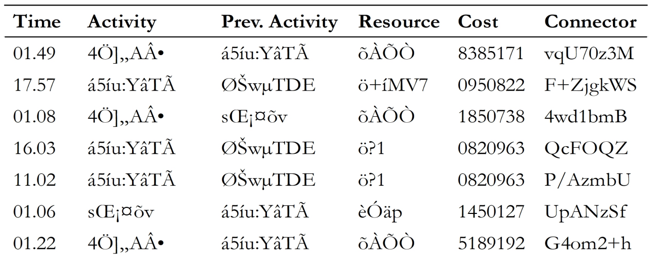
Encrypting The Plain Part. In this step, activities are encrypted by a deterministic encryption method like AES. A deterministic encryption method has to be used because for discovering a DFG or a process model, differences should be preserved. Fig. 7 shows the result after encrypting activities.

However, after encrypting, detecting "START" activities seem to be impossible, and without detecting them, extracting the relations is not possible. For identifying the "START" activities, we can go through the "Activity" and "Prev. Activity" columns, the activities which are appeared in the "Prev. Activity" column but not appeared in the "Activity" column are the "START" activities.

Fortifying Encryption and/or Projecting Event Logs. In our sample, resources are encrypted by a deterministic encryption method (AES-ECB), and costs are encrypted by homomorphic encryption, which preserves differences. Consequently, by comparison, one can find the minimum and maximum cost, which can be used as knowledge for extracting confidential information (e.g. name of resource). In order to decrease the effect of such analyses, fortifying encryption and/or projecting event logs could be done. E.g., resources can be encrypted by a probabilistic encryption (e.g. AES-CTR), and costs can be removed. In fact, all attributes not needed for process discovery can be removed.

6 Evaluation

We consider three evaluation criteria for the proposed approach while performance is also taken into account.



Time	Activity	Prev. Activity	Resource	Cost	Connector
01.49	4Ö],,AÂ•	á5íu:YâTÃ	õÀÕÒ	8385171	vqU70z3M
17.57	á5íu:YâTÃ	ØŠwµTDE	ö+íMV7	0950822	F+ZjgkWS
01.08	4Ö],,AÂ•	sŒ¡¤õv	õÀÕÒ	1850738	4wd1bmB
16.03	á5íu:YâTÃ	ØŠwµTDE	ö?1	0820963	QcFOQZ
11.02	á5íu:YâTÃ	ØŠwµTDE	ö?1	0820963	P/AzmbU
01.06	sŒ¡¤õv	á5íu:YâTÃ	èÓäp	1450127	UpANzSf
01.22	4Ö],,AÂ•	á5íu:YâTÃ	õÀÕÒ	5189192	G4om2+h

Fig. 7: The event log after encrypting activities

- *Ensuring Confidentiality:* as explained in Section 5, we can increase the level of confidentiality by defining different environments an indicating level of information which is accessible by each environment. In addition, using multiple encryption methods and our connector method for disassociating events from their cases improve confidentiality.
- *Providing Reversibility:* when the keys and the value used for making times relative are given, both ICS and ECS are reversible, which means that transparency is addressed by the proposed approach.
- *Proving Accuracy:* to prove the accuracy of our approach, by a case study we show that DFG of the original event log (AEL) and DFG of the secure event logs (i.e., AEL' and AEL'') are the same, and consequently corresponding process models are similar.

6.1 Proving Accuracy

As can be seen in Fig. 1, to prove accuracy, we need to show that the abstraction of the original event log is the same as the abstraction of the internal event log ($AEL = AEL'$) (rule (1)), and also the abstraction of the internal event log is the same as the abstraction of the external event log, which is encrypted ($AEL' = ECS^{-1}(AEL'')$) (rule (2)). For this purpose, we have implemented four plugins for ProM including; "ICS", "ECS", "DFG from secure logs", and "DFG from regular logs". "ICS" is used for converting an event log in regular XES format to the internal version of secure event log, "ECS" is used for converting internal version of secure event log to the external version of secure event log, "DFG creator from secure logs" is able to make a DFG based on the secure version of event log, and "DFG creator from regular logs" is used to make a DFG from regular XES log. These plugins have been used along with a case study of real life logs to prove the accuracy. In summary:

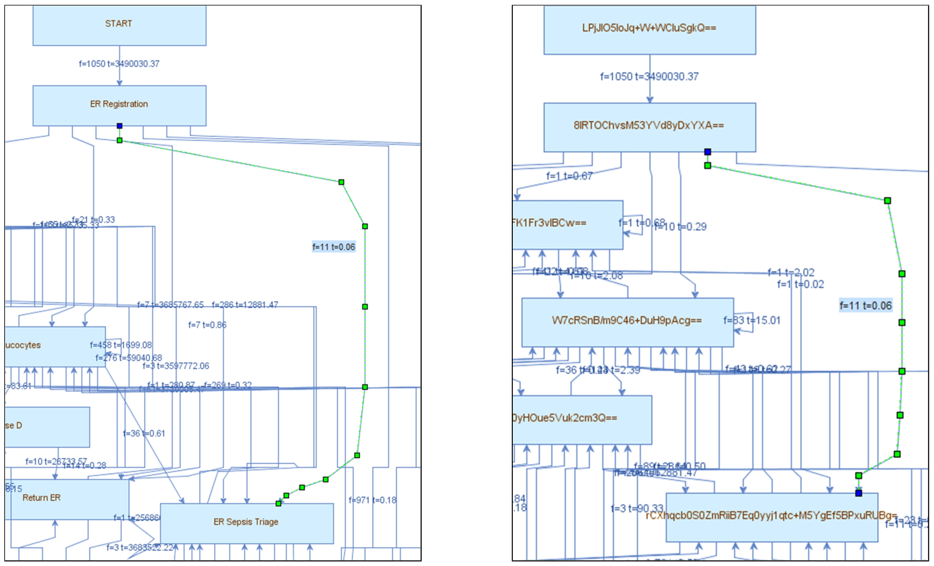
$$AEL = AEL' \Rightarrow PM = PM' \tag{1}$$

$$AEL' = ECS^{-1}(AEL'') \Rightarrow PM' \approx ECS^{-1}(PM'') \tag{2}$$

6.2 Case Study: Real Life Log of Sepsis Patients

The real-life event log for a group of sepsis patients in a hospital [15], containing 1050 cases, 15214 events, and 16 event classes, is used to prove the accuracy.

In the first step, EL', and EL'' have been created by "ICS", and "ECS" plugins respectively. Then, to verify that AEL is identical to AEL', "DFG from Regular Logs" and "DFG from Secure Logs" have been used to produce corresponding DFGs. The resulting DFGs were exactly the same. Because of the space limitations, we are not able to show them. Finally, to prove that AEL' is the same as AEL'', where activities are encrypted, we have used "DFG from Secure Logs" plugin. To be able to take a closer look at the AEL' and AEL'', in Fig. 8, we have zoomed in both of them and highlighted a specific path from



(a) A part of the plain DFG (b) A part of the encrypted DFG

Fig. 8: Comparing DFG from EL' with DFG from EL'': both graphs are identical, modulo renaming and layout differences

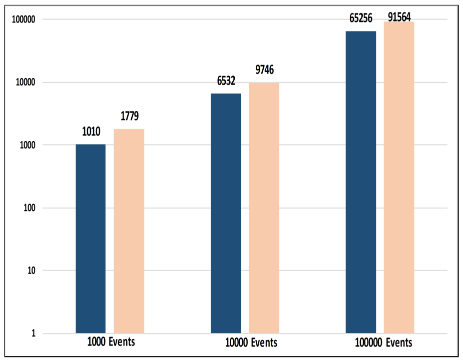
"ER Registration" to "ER Sepsis Triage". As can be seen in Fig. 8, both AEL' and AEL'' show the same relation between these two activities. The frequency of this relation is 11 and the average time is 0.06 (f=11, t=0.06). In addition, this figure shows that "ER Registration" has no real input link, and "ER Sepsis Triage" has ten input links and eight output links.

6.3 Performance

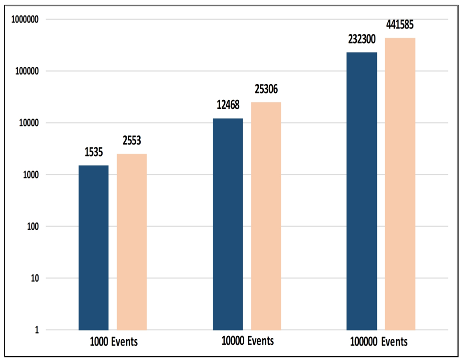
Fig. 9 shows how the application scales when using benchmarking event logs [3] and increasing the number of events exponentially. All runtimes have been tested using an Intel i7 Processor with 1.8GHz and 16 GB RAM. The darker bars show the execution time of the "DFG from regular logs", and the lighter bars show the execution time of the "DFG from the secure logs". We see a linear increase of the runtime in milliseconds when adding choices or loops.

7 Conclusions

This paper presented a novel approach to ensure confidentiality in process mining. We demonstrated that confidentiality in process mining can not be achieved by only encrypting the whole event log. We discussed the few related works, most of which use just encryption, and explained their weaknesses. Moreover,



(a) Exec. time for choice loop events



(b) Exec. time for sequence loop events

Fig. 9: Scaling the program to larger event logs

we elaborated on the open challenges in this research area. The new approach is introduced based on the fact that there always exist a trade-off between confidentiality and data utility. Therefore, we reasoned backwards from the desired results and how they can be obtained with as little data as possible.

Here, the desired result was a process model and the solution presented by introducing a framework for confidentiality that can be extended to include other forms of process mining, e.g., conformance checking, performance analysis, social network analysis, etc. (i.e., different ICS and ECS could be explored for different process mining activities). A new method named "Connector" has been introduced, which can be employed in any situation in which we need to store some associations securely. For evaluating the proposed approach, four plugins have been implemented and a real-life log was used as a case study. The approach is tailored towards the discovery of the directly follows graph. Also, the framework could be utilized in cross-organizational context such that each environment could cover specific constraints and authorizations of a party.

References

1. Van der Aalst, W.: Process mining: data science in action. Springer (2016)
2. Van der Aalst, W.: Responsible data science: using event data in a "people friendly" manner. In: International Conference on Enterprise Information Systems. pp. 3–28. Springer (2016)
3. Van der Aalst, W.: Benchmarking logs to test scalability of process discovery algorithms. Eindhoven University of Technology. `https://data.4tu.nl/repository/uuid:1cc41f8a-3557-499a-8b34-880c1251bd6e` (2017), [Online; accessed 17-September-2018]
4. Van der Aalst, W.: Process discovery from event data: Relating models and logs through abstractions. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **8**(3), e1244 (2018)

5. Van der Aalst, W., Adriansyah, A., De Medeiros, A.K.A., Arcieri, F., Baier, T., Blickle, T., Bose, J.C., Van Den Brand, P., Brandtjen, R., Buijs, J., et al.: Process mining manifesto. In: International Conference on Business Process Management. pp. 169–194. Springer (2011)
6. Van der Aalst, W., Bichler, M., Heinzl, A.: Responsible data science. Business & Information Systems Engineering **59**(5), 311–313 (Oct 2017)
7. Bellare, M., Rogaway, P.: Introduction to modern cryptography. Ucsd Cse **207**, 207 (2005)
8. Burattin, A., Conti, M., Turato, D.: Toward an anonymous process mining. In: Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on. pp. 58–63. IEEE (2015)
9. Kapoor, V., Poncelet, P., Trousset, F., Teisseire, M.: Privacy preserving sequential pattern mining in distributed databases. In: Proceedings of the 15th ACM international conference on Information and knowledge management. pp. 758–767. ACM (2006)
10. Katz, J., Menezes, A.J., Van Oorschot, P.C., Vanstone, S.A.: Handbook of applied cryptography. CRC press (1996)
11. Leemans, M., Van der Aalst, W., van den Brand, M.G.: Hierarchical performance analysis for process mining. In: Proceedings of the 2018 International Conference on Software and System Process. pp. 96–105. ACM (2018)
12. Leemans, S.J., Fahland, D., Van der Aalst, W.: Scalable process discovery and conformance checking. Software & Systems Modeling **17**(2), 599–631 (2018)
13. Liu, C., Duan, H., Qingtian, Z., Zhou, M., Lu, F., Cheng, J.: Towards comprehensive support for privacy preservation cross-organization business process mining. IEEE Transactions on Services Computing (1), 1–1 (2016)
14. Ma, C.Y., Yau, D.K., Yip, N.K., Rao, N.S.: Privacy vulnerability of published anonymous mobility traces. IEEE/ACM transactions on networking (TON) **21**(3), 720–733 (2013)
15. Mannhardt, F.: Sepsis Cases - Event Log. Eindhoven University of Technology. `https://doi.org/10.4121/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460` (2016), [Online; accessed 17-September-2018]
16. Mannhardt, F., de Leoni, M., Reijers, H.A., Van der Aalst, W., Toussaint, P.J.: Guided process discovery–a pattern-based approach. Information Systems **76**, 1–18 (2018)
17. Rozinat, Günther, C.W.: Privacy, security and ethics in process mining. `http://coda.fluxicon.com/assets/downloads/Articles/PMNews/Privacy-Security-and-Ethics-In-Process-Mining.pdf` (2016), [Online; accessed 17-September-2018]
18. Sani, M.F., van Zelst, S.J., Van der Aalst, W.: Repairing outlier behaviour in event logs. In: International Conference on Business Information Systems. pp. 115–131. Springer (2018)
19. Tillem, G., Erkin, Z., Lagendijk, R.L.: Privacy-preserving alpha algorithm for software analysis. In: 37th WIC Symposium on Information Theory in the Benelux/6th WIC/IEEE SP Symposium on Information Theory and Signal Processing in the Benelux
20. Zhan, J.Z., Chang, L., Matwin, S.: Privacy-preserving collaborative sequential pattern mining. Tech. rep., Ottawa Univ (Ontario) School of Information Technology (2004)